

Benchmarking News Recommendations in a Living Lab

Frank Hopfgartner¹, Benjamin Kille¹, Andreas Lommatzsch¹, Till Plumbaum¹,
Torben Brodt², and Tobias Heintz²

¹ Technische Universität Berlin, Ernst-Reuter-Platz 7, 10587 Berlin, Germany

² plista GmbH, Torstraße 33–35, 10119 Berlin, Germany

Abstract. Most user-centric studies of information access systems in literature suffer from unrealistic settings or limited numbers of users who participate in the study. In order to address this issue, the idea of a living lab has been promoted. Living labs allow us to evaluate research hypotheses using a large number of users who satisfy their information need in a real context. In this paper, we introduce a living lab on news recommendation in real time. The living lab has first been organized as News Recommendation Challenge at ACM RecSys’13 and then as campaign-style evaluation lab NEWSREEL at CLEF’14. Within this lab, researchers were asked to provide news article recommendations to millions of users in real time. Different from user studies which have been performed in a laboratory, these users are following their own agenda. Consequently, laboratory bias on their behavior can be neglected. We outline the living lab scenario and the experimental setup of the two benchmarking events. We argue that the living lab can serve as reference point for the implementation of living labs for the evaluation of information access systems.

1 Introduction

Over the years, significant effort has been done to establish appropriate measures, frameworks, and datasets that allow for a fair and unbiased evaluation of novel approaches for information retrieval and recommender systems, also referred to as information access systems. In the field of information retrieval, consortia such as TREC, CLEF and FIRE provided the ground for focused research on various aspects of information retrieval. In the field of recommender systems, the release of the Netflix dataset and the associated challenge was a key event that led to an advance of research on recommender systems. Although the release of common datasets was of great benefit for the research community, focusing on them does not come without drawbacks [34]. While datasets can be used to fine-tune models and algorithms to increase precision and recall even further, the user is often kept out of the loop [20, 6]. However, the user plays an essential role in the evaluation of information access systems. It is the user’s information need that needs to be satisfied and it is the user’s personal interests that need to be considered when adapting retrieval results when providing good recommendations.

Consequently, user-centric evaluation of information access systems is essential to evaluate the full performance of such systems. Unfortunately though, most researchers often have limited access to real user interactions that would allow testing research hypotheses in a large scale. In order to address this issue, the application of a living lab has been proposed (e.g., [19, 20]) that grant researchers access to real users who follow their own information seeking tasks in a natural and thus realistic contextual setting. For user-centric research on information access systems, realistic context is essential since it is a requirement for a fair and unbiased evaluation.

In this paper, we introduce a living lab for the real-time evaluation of news recommendation algorithms. The lab infrastructure was used during the News Recommender Systems (NRS) challenge which was held in conjunction with ACM RecSys 2013 and during the campaign-style lab NEWSREEL of CLEF 2014. By participating in this living lab, participants were given the opportunity to develop news recommendation algorithms and have them tested by potentially millions of users over a longer period of time. The task which is addressed within this living lab is to provide recommendations under the typical restrictions (e.g., time constraints) of real-world recommender systems. Such restrictions pose requirements regarding scalability as well as complexity for the recommendation algorithms. We introduce this living lab scenario and describe two benchmarking events that show how the living lab can be used to promote research in the news recommendation domain.

This paper is organized as follows. In Section 2, we provide an overview of related work on the evaluation of information access systems. Section 3 introduces the specific domain of providing recommendations on news portals. Section 4 introduces the setup and infrastructure of the living lab for the evaluation of such algorithms. Two benchmarking events where the living lab has been applied are outlined in Section 5. Section 6 concludes this paper.

2 Evaluation of Information Access Systems

One of the main prerequisites of modern research is the design and implementation of appropriate evaluation protocols which allow us to compare novel techniques with existing state-of-the-art approaches. In the information retrieval domain, the origin of such protocol is based on the early work of Cleverdon et al. [9] who introduced the idea of evaluating algorithms in a controlled setting using a test dataset. Thanks to the implementation of the Text REtrieval Conference (TREC) initiative [34], the use of test datasets, consisting of document collections, pre-defined search tasks and relevance assessments has become the de-facto evaluation protocol for IR research. Over the years, various datasets from different domains have been published that promoted research on information access systems significantly. In the context of recommender systems evaluation, these domains include books, music, jokes, movies and many others [7, 11, 14, 35].

Although this evaluation paradigm helped us to study multiple research challenges in the field, it did not come without drawbacks. Clough and Sanderson

[10] point out that the main limitations include the artificial nature of the setting that is defined within this batch evaluation and the negligence of the user and their role in the information gathering task. Similar issues have been observed in the evaluation of recommender systems. Konstan and Riedl [22] argue that recommender systems' evaluation should consider the user experience rather than rating prediction accuracy. Additionally, they note that other factors such as scalability, diversity, and novelty play an important role. They propose to define more sophisticated quality measures to capture user experience. Shani and Gunawardana [30] discuss recommender systems evaluation in three settings: (i) experiments on data sets, (ii) user studies, and (iii) online evaluation interacting with actual users. Herein, they state that online evaluation provides the strongest evidence on how well a recommender systems performs. Neither user studies nor experiments on data sets achieve similar expressiveness.

In order to address these limitations, two approaches have been proposed: (1) The extension of test collections by adding user interaction records (e.g., within the TREC Interactive track [12] and the HARD track [2]) and (2) the simulation of user interaction [17, 18] that allow to run batch evaluation without the constant requirement of user input.

Both methods come with their own limitations: While bringing the user into the loop can be considered to be a step in the right direction, large user bases are required to confirm research hypotheses [6]. However, this often is not an issue for commercial providers of information access systems. Therefore, having large user bases, user-centric online evaluation is the first choice for the evaluation of such systems. A guideline for large-scale online testing of recommender systems, also referred to as A/B testing, is provided by Amatriain [3]. In order to test improvements or variants of information access systems, new instances of these systems are released that often differ in one key aspect from the original system only. These instances are referred to as System A and System B. Users of the system are then split into different groups: Group A and Group B. When users of Group A want to access the system, they are forwarded to System A. Users of Group B, on the other hand, are forwarded to System B. Observing the users' interactions and their behavior over time, conclusions can be drawn on which of these systems is better.

Although the protocol is rather simple, it comes with a major drawback. In order to get meaningful results, a large user base is required. While this is no problem for commercial providers, the lack of access to actual users hinders non-commercial research significantly. At the SIGIR 2009 workshop on Future Information Retrieval Evaluation [19], participants promoted the application of a living lab to address this issue. Pirolli [27] argues that such living labs could attract researchers from many different domains. Kelly et al. [20] promotes the role of a living lab and its advantages as follows:

A living laboratory on the Web that brings researchers and searchers together is needed to facilitate ISSS [Information-Seeking Support System] evaluation. Such a lab might contain resources and tools for evaluation as

well as infrastructure for collaborative studies. It might also function as a point of contact with those interested in participating in ISSS studies.

A first proposal for a living lab for information retrieval research is outlined by Azzopardi and Balog [4]. They propose a generic infrastructure for such lab which allows different parties (i.e., researchers, commercial organizations, evaluation forums, and users) to communicate with each other. Moreover, they illustrate how this infrastructure can be used in a specific use case. Although their work can be considered to be a key contribution for the definition of living labs, their work remains theoretical. In this paper, we introduce the application of a living lab for the benchmarking of news recommendation algorithms in real time. Within this living lab, different parties interact with each other using a shared infrastructure: Users visit news portals of commercial providers, these visits are reported to researchers whose task is to identify other news articles of this provider which are then recommended to the user for further reading. To the best of our knowledge, it is the first living lab for the evaluation of information access systems. In the next sections, we first introduce the use case of news article recommendation, followed by an overview of the living lab setup.

3 Real-Time News Recommendation

Real-time news recommendation differs from the most *traditional* recommender scenarios which have been studied in literature. Instead of computing recommendations based on a static set of users and items, the challenge here is to provide recommendations for a news article stream characterized by a continuously changing set of users and items. The short lifecycle of items and the strict time-constraints for recommending news articles make great demands on the recommender strategies. In a stream-based scenario the recommender algorithms must be able to cope with lot of newly created articles and should be able to discard old articles, since recommended news articles should be “new”. Thus, the recommender algorithms must be steadily adapted to meet the special requirements of the news recommendation scenario. Moreover, the recommendations have to be provided fast since most users are not willing to wait for recommendations that they did not even request in the first place. In order to clarify the types of recommendations which are possible, we outline in this section typical recommendation methods that are able to provide recommendations within a very short period of time, namely: (1) Most recently read articles, (2) Most popular articles, (3) User-based collaborative filtering (CF), (4) Item-based collaborative filtering, and (5) textual similarity of the news article descriptions.

The basic idea of a recommender of most recently read articles is that those articles which are currently read by the community are the most relevant articles for a potential visitor of a news portal. A similar idea is presented by Phelan et al. [26] who use most recent tweets to recommend real-time topical news. The strength of this recommender is that it has a low computational complexity. It provides recommendations very fast and scales well with the number of requests.

Since this algorithm considers neither any contextual feature nor individual user preferences, the recommendation precision is limited. In other words, the recommendations do not reflect the user’s profile and are not optimized to the service context.

News articles most frequently requested by the community are typically interesting for most of the users (e.g., [33]). A most popular recommender counts the number of requests per article and suggests the most popular articles still unknown to the user. The strengths of the approach are that the algorithm is simple and provides results having a high probability to be relevant. A weakness of the most popular recommender is that it does not consider individual user preferences and it does not recommend breaking news articles (due to the fact that it takes time for an article to get a large number of impression events). The recommendations are neither personalized nor context-aware.

User-based collaborative filtering (e.g., [15, 1]) is the most popular approach in the recommender domain. In order to compute suggestions, this recommender determines similar users based on the accessed items (e.g., news articles). News portals are typically dynamic systems characterized by a large number of article creates and user-article interactions. The advantages of this recommendation approach are that it considers the user preferences and provides personalized results. Disadvantages are that storing the user-item interaction is resource demanding and computing similar users is computationally expensive. In addition, collaborative filtering approaches suffer from the “cold-start” problem, making it challenging to compute high-quality recommendations for new users.

Similar to user-based collaborative filtering, item-based collaborative filtering techniques (e.g., [29]) can suggest news articles read by the same users that also read the current news article. In contrast to user-based CF, item-based CF recommenders are robust against noisy user IDs. Additionally, item-based recommendations are often also related on a content-level, due to the observation that users are interested in content-based categories (e.g., basketball). The strength of an item-based collaborative filtering recommender is that this algorithm provides highly relevant suggestions for the documents the user read in the past. The algorithm is robust against noisy user IDs and computes recommendations based on the wisdom of the crowd. Weaknesses are that the algorithm does not consider the context. Additionally, an item-based collaborative filtering approach cannot provide good recommendations for new items having only a small number of ratings (“cold-start problem”).

Another approach to provide recommendations is to determine content-based similarity (e.g., [25]) between news articles. The strength of a content-based recommender is that it does not require user feedback and can recommend completely new articles. The disadvantage is that the content does neither say much about the article’s quality nor whether the article matches the individual user preferences. The processing of natural language texts and the extraction of the most relevant terms is computationally expensive and requires robust linguistic tools. As discussed by several researchers (e.g., [24]), content-based features have a much lower impact on the items’ relevance than collaborative features.

Each of the presented methods has its specific strength and weaknesses. The most recently read recommender and the most popular recommender tend to suggest news articles which are of interest for most of the users, but do not consider the individual user preferences. Since both algorithms have a low resource demand, these algorithms can efficiently handle a large number of requests. User-based collaborative filtering provides personalized suggestions based on the preferences of similar users. In contrast to a most popular recommender this algorithm has a higher computational complexity since the preferences of all users must be managed in order to determine the most similar users. Item-based collaborative filtering algorithms as well as algorithms suggesting news articles based on the textual similarity of news articles recommend items related to the currently viewed news article. Both algorithms suggest news articles related to the currently requested article helping the user to track the development of a story and to discover news articles similar with regards to contents.

One approach to overcome the disadvantages of all approaches is to combine the algorithms in a recommender ensemble that can automatically identify the best performing algorithms for a specific domain and adapt its recommendation technique accordingly over time. Lommatzsch [23] analyzes news recommender quality dependent of the domain and the context to find out what approach works best for which type of request. Benchmarking different news recommendation algorithms, he observes that there is not *one* optimal algorithm that outperforms all other recommendation strategies. Therefore, he concludes that the recommendation performance depends on context and domain.

4 Living Lab Scenario

As argued above, the aim of a living lab is to bring together users and researchers, e.g., by providing an infrastructure that allows researchers to test algorithms and systems under real-life conditions. In this living lab, researchers can benchmark news recommendation techniques in real-time by recommending news articles to actual users that visit commercial news portals to satisfy their individual information needs, i.e., participants are facing real users in a living lab environment. In Section 4.1, we first introduce the domain of online news recommendation in detail. Section 4.2 provides an overview of the publishers and the user base, i.e., the content and the target group that is relevant for this scenario. The infrastructure is introduced in Section 4.3.

4.1 Online News Recommendation: The plista use case

Many online news portals display on the bottom of their articles a small widget box labelled “You might also be interested in”, “Recommended articles”, or similar where users can find a list of recommended news articles. Dependent on the actual content provider, these recommendations often consist of a small picture and accompanying text snippets. Figure 1 illustrates the typical position of the recommendations on a typical news portal page.

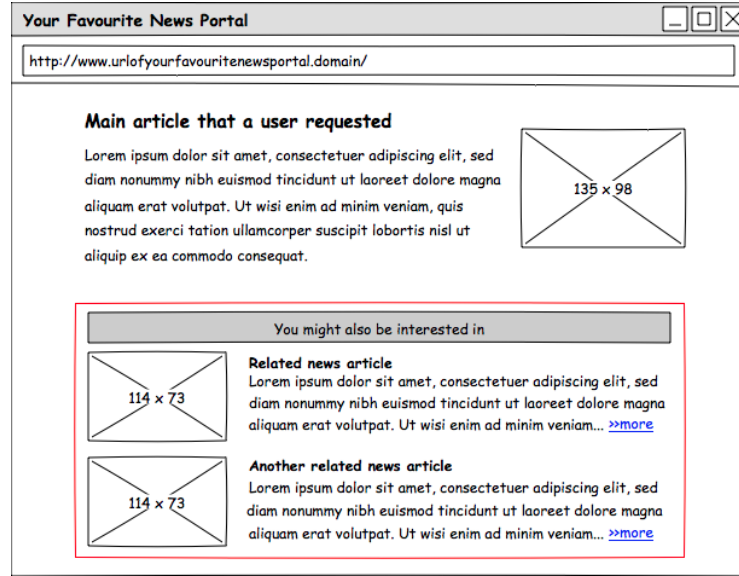


Fig. 1. Common position of the recommended news articles on a news portal.

While some publishers provide their own recommendations, more and more providers rely on the expertise of external companies such as plista³, a data-driven media company which provides content and advertising recommendations for thousands of premium websites (e.g., news portals, entertainment portals). Whenever a user reads an article on one of their customers' web portals, the plista service provides a list of related articles. In order to outsource this recommendation task to plista, the publishers firstly have to inform them about newly created articles and updates on already existing articles on their news portal. In addition, whenever a user visits one of these online articles, the content provider forwards this request to plista. These clicks on articles are also referred to as impressions. Plista determines related articles which are then forwarded to the user and displayed in above mentioned widget box as recommendations. Having a large customer base, plista processes millions of user visits in real time on a daily basis. By setting up this living lab, plista grants research teams access to a certain amount of these requests in order to promote research on real-time news article recommendation. An overview of the publishers and users that are relevant for this scenario is provided in the next section.

4.2 Publishers and Users

Due to plista's business focus on the German-speaking market in Central Europe, the main target group for their recommendations are German-speaking people.

³ <http://www.plista.com/>

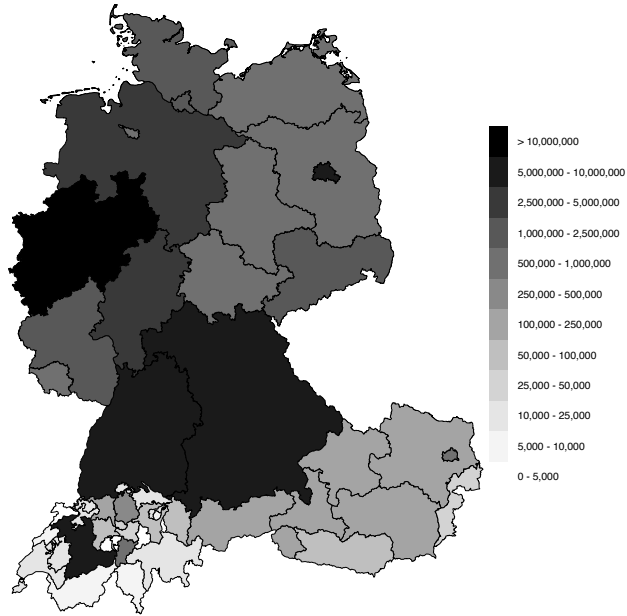


Fig. 2. First-level and second-level NUTS in Germany, Austria, and Switzerland from where requests for articles were triggered. The scale indicates the number of requests.

German is the most widely spoken mother tongue in the European Union and understood by 30% of all EU citizens [32]. As of June 2014, it is the second most used language in the internet⁴, indicating the strong role that the internet plays as information source for the target group. A shared language, historical ties and geographic proximity provide the ground for an intensive cultural exchange between the largest German-speaking countries Germany, Austria and Switzerland. This is also reflected in the digital media landscape. With all three countries ranked amongst the Top 15 countries on the 2014 World Press Freedom index, publishers of these countries are able to publish articles on their portals without larger fear of political consequences. A multitude of online publishers exist in these countries that focus on daily news on a regional, national or international level, or on specific domains such as sports, business or technology. Thousands of them rely on plista to provide recommendations for their visitors. In the context of this living lab, plista forwards the requests of a diverse selection of these clients, including regional and local news publishers, as well as domain-centric portals. An analysis of a four-week log file dump of activity records for selected domains (see Section 5.1 for further details) reveals that 81.8% of all requests for websites were requested from visitors from Germany,

⁴ According to http://w3techs.com/technologies/overview/content_language/all, accessed on 19 June 2014.

Austria, or Switzerland. Figure 2 highlights the regions from where these requests were triggered. Figure 3 visualizes which devices (i.e., tablets, phones, desktop computers, crawlers, or bots) were used to access the news portals. We interpret the changing proportions over time as an indication that both time of access and the choice of device is decided by the users. In other words, users were accessing the sites following their own personal agendas. A preliminary analysis of users’ behavior is performed by Esiyok et al. [13] and Said et al. [28].

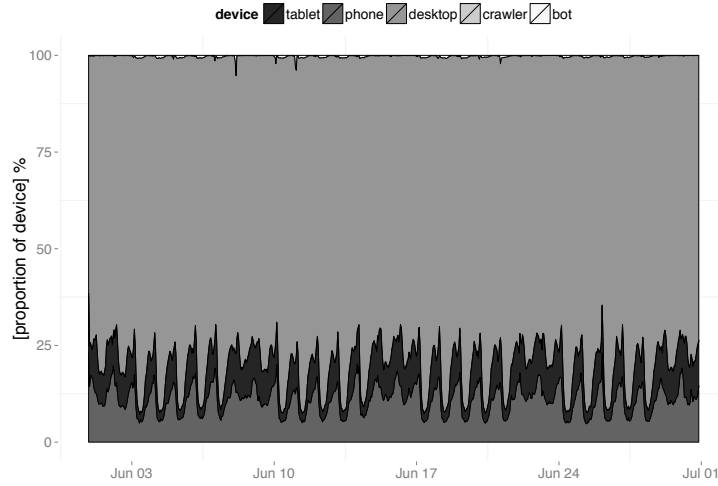


Fig. 3. Distribution of devices used to access news portals.

Concluding, we argue that the average users of this living lab are German-speaking Europeans who follow their own information need on a diverse set of news portals. How researchers can evaluate their algorithms for these news portals and their visitors is outlined in the next section.

4.3 Infrastructure

As described above, access to the publishers and users is provided by plista, who created an API for researchers that allows them to benchmark news recommendation algorithms and have them tested by a subset of their customers’ visitors. The infrastructure that is required for this living lab has been developed in the context of the research project EPEN⁵. Figure 4 visualizes the data flow between the different players of this living lab, namely the visitors of news portals, the news portals, the Open Recommendation Platform (ORP) [8], and the servers of the individual participants who benchmark their algorithms.

⁵ <http://www.dai-labor.de/en/irml/epen/>

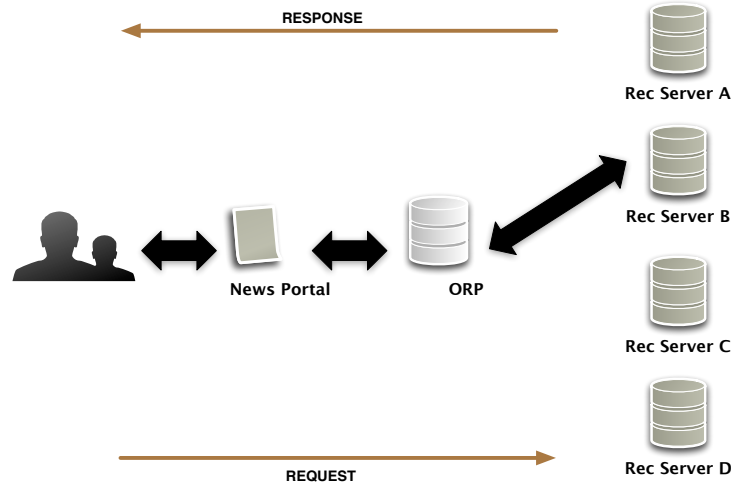


Fig. 4. Data flow within the living lab.

The Open Recommendation Platform (ORP) [8] is the core platform of the living lab since it handles the communication between the participants and plista. ORP receives recommendation requests from visitors from various websites offering news articles and forwards the incoming requests to registered researchers. The platform is capable of delivering different recommendation implementations and of tracking the recommender results.

After registering using the graphical user interface of the platform, researchers first need to provide a server address on which their implementation of a news recommender is running. Moreover, they can register different algorithms that can simultaneously be run. Once registered, ORP will send HTTP POST requests, including item updates, event notifications and recommendation requests to this server. Event notifications are the actual user interactions, i.e., users' visits, referred to as impressions, to one of the news portals that rely on the plista service, or clicks to one of the recommended articles. The item updates include information about the creation of new pages on the content providers server and it allows participants to provide content-based recommendations. Recommender algorithms and evaluation models can also be build on top of the context, which includes the user id provided by a cookie, publisher id, browser, device, operating system and more, either from the http context or additionally being enhanced by plista using categorization heuristics and classifiers. Expected responses to the recommendation requests are related news articles from the same content provider, which are then provided as recommendations to the visitors of the page. Since recommendations need to be provided in real-time, the expected response has to be send within 100ms, i.e., recommenders have to be quick. If too much time is lost due to network latency (e.g., when the participant has a slow internet connection or is physically remote from the ORP server), the algorithms can also

be installed on a server provided by plista. Hence, the participants experience typical restrictions for real-world recommender systems such as scalability and complexity of the recommendation algorithms.

When participating in the living lab, participants have the chance to continuously update their parameter settings in order to improve their performance levels. Therefore, the ORP visualizes the algorithms' performances over time. An example is shown in Figure 5. Performance is measured in impressions, clicks and click-through rate (CTR) per day. An impression record is created whenever a user reads an article and the participant received a request to provide recommendations for this visit. Clicks represent users following links to articles that have been recommended while reading a news article. CTR is defined as the ratio of clicks over impressions.

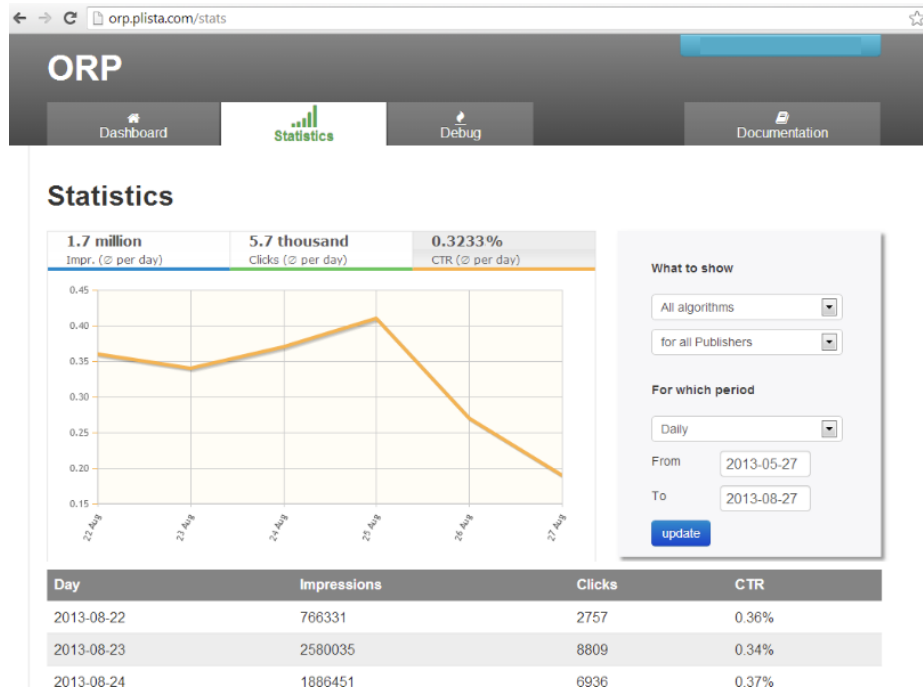


Fig. 5. Screenshot of the ORP.

5 Evaluation Scenarios

So far, the living lab infrastructure and related datasets have been used in two evaluation and benchmarking campaigns, namely in the News Recommendation

Challenge (NRS'13), held in conjunction with ACM RecSys 2013 and in NEWS-REEL, a campaign-style evaluation lab of CLEF 2014. In the remainder of this section, we outline the experimental setup of these two events.

5.1 The News Recommendation Challenge 2013

The living lab was first introduced to the research community in 2013, when we organized a workshop and challenge on news recommendation systems (NRS) [31] in conjunction with ACM RecSys 2013. The aim of this workshop was to bring together researchers and practitioners around the topics of designing and evaluating novel news recommender systems. Additionally, the aim of the challenge was to allow participants to evaluate their method by directly interacting with real-world news recommender systems. The challenge featured a data set designed to bootstrap a news recommender system and access to the living lab for a few weeks. During the last two weeks leading up to the conference, each participant's system performance was measured with respect to the ratio of clicks per recommendation request. The two phases of the challenge are outlined in the remainder of this section.

Phase 1: Training. In the first stage, a log file dump of the activity records that plista processed in June 2013 for recommending news articles in real-time was provided. While plista provides this service for thousands of online portals, this dataset contains records for a limited number of news portals, covering different spectra of the news world such as general, sports-related, or information technology related news. As mentioned above, plista's domestic market is Central Europe. Therefore, all news providers publish articles in German.

The corpus consists of four types of activities that have been performed by two types of actors on selected online domains: Adding and updating articles (done by the online editors of the respective news portal) as well as reading an article and clicking on a recommendation (the latter two activities being performed by the online customer, i.e., the readers of the online portals). Figure 6 visualizes the number of impressions over time for an exemplary news domain. The dataset allowed participants to tune their recommendation algorithms before the actual real-time challenge commenced. For a more detailed description of the dataset, the reader is referred to [21].

Phase 2: Benchmarking in Living Lab. In the second stage, participants were asked to provide recommendations in real-time for actual users. After registering with the Open Recommendation Platform, the participants received updates for ten publishers and requests for recommendations triggered by the visitors of these news portals. For a period of two weeks, we recorded the number of clicks, the number of requests and the click-through rate of all participating recommenders.

Eight teams participated in the challenge who could submit a multitude of recommenders. Overall, we counted 23 algorithms that competed against each

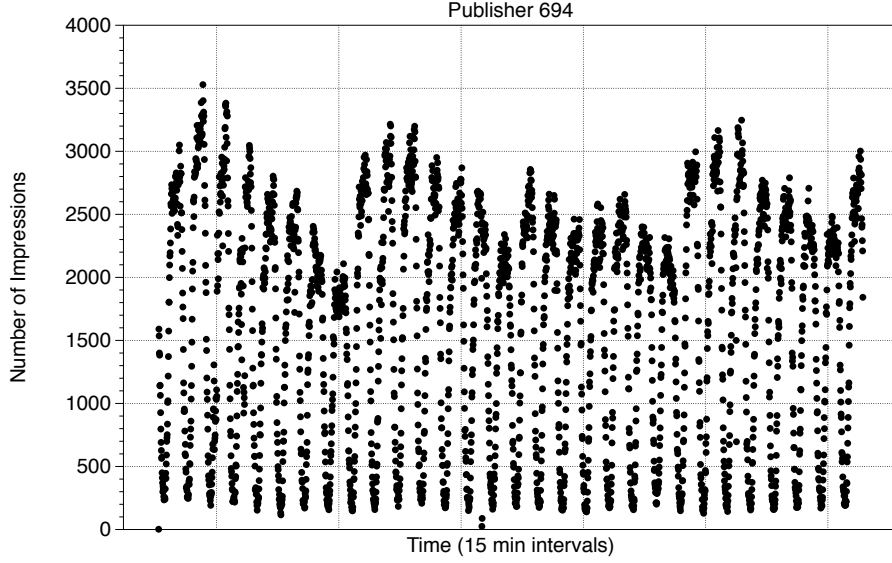


Fig. 6. Recorded impressions over time for an exemplary news domain. Each dot corresponds to the number of interactions within a 15 minute time interval. The whole time frame corresponds to a full month of interactions.

other and against four baseline runs. For further details about the baseline algorithms, the reader is referred to [23]. Since the main focus of this paper is to outline the living lab scenario, we will only *briefly* discuss the results of the challenge. We could observe a much larger number of requests in the challenge’s early stages. This resulted from more and more teams joining the challenge reducing the traffic routed to individual algorithms. Moreover, we noticed that the click-through rates started relatively low and increased with \approx factor 2 after 3 days. We also observed that the performance of the recommenders share similarities. On Day 4, for example, all baseline recommenders reported a local high, followed by a local low on Day 6. Similar patterns could be observed on Days 11, 12, and 13. Several tens of thousands recommendations had been submitted until the organizers announced the winners thus concluding the challenge. Since the main focus of this paper is to introduce the evaluation setting of this living lab, a detailed discussion on these effects is out of scope.

5.2 CLEF NEWSREEL 2014

Building on the experiences we gained when organizing the NRS challenge, we revised the experimental setup to promote further research on the offline and online evaluation of news recommendation algorithms. In 2014, the introduced

infrastructure and dataset was used to organize NEWSREEL⁶, a campaign-style evaluation lab of CLEF 2014. NEWSREEL consisted of two tasks that are outlined in the remainder of this section. Note that this section provides an overview of the tasks only. For an overview of the participating teams and their performances, the reader is referred to the lab overview paper in the working notes proceedings of CLEF'14.

Task 1: Predict interactions in an offline dataset. Due to the organization of the Netflix challenge, evaluation of recommendation algorithms is dominated by offline evaluation scenarios. Addressing this evaluation scenario, we re-used a subset of the above mentioned dataset [21] to allow for an offline evaluation. The subset consisted of all updates and interactions of ten domains, focusing on local news, sports, business, technology, and national news, respectively. Before releasing the dataset, we identified fifteen time slots of 2–6 hours of length for each of the ten domains and removed all (user, item)-pairs within these slots. The task was to predict the interactions that occurred during these time periods. Note that the complete dataset had already been released during the NRS challenge. Participants were therefore advised that they must not use this dataset for this prediction task. Predictions were considered successful if the predicted (user, item)-pair actually occurred in the data set. In the evaluation, all partitions were treated separately, the winning contribution was determined by aggregating the results from the individual partitions. Participants were not asked to provide running code, but instead had to provide files with their predictions.

Task 2: Recommend news articles in real-time. In the second task, participants got the chance to benchmark recommendation algorithms in the living lab. Lab registration started in November 2013 and closed in May 2014. Once registered for CLEF, participating teams received an account on the Open Recommendation Platform. After providing a server address and after registering an algorithm in the dashboard, they were constantly receiving requests for recommendations. The platform was constantly online, thus leaving the participants various months time to fine-tune their algorithms. In order to compare the different participating teams, we defined three evaluation periods of two weeks duration each during which we recorded the numbers of clicks, numbers of requests and the click-through rate (CTR). The evaluation periods were scheduled in early February 2014, early April 2014 and late May 2014.

5.3 Discussion

The two benchmarking events that we presented allowed researchers to run news recommendation algorithms under real conditions and have them tested by a large number of users. As common in living labs, the users were not controlled, i.e., they were following their own agenda while browsing the different news

⁶ <http://www.clef-newsreel.org/>

portals. When setting up a living lab, various issues need to be considered, including legal and ethical issues (e.g., protection of users’ privacy and intellectual property and handling of sensitive user interaction streams), but also technical and practical challenges (e.g., setting up and maintaining the lab infrastructure and the definition of evaluation scenarios). In the remainder of this section, we outline how we addressed these issues.

Data protection is a key requirement for running the living lab. The lab infrastructure is provided by *plista*, a company based in Berlin, Germany. Therefore, they are subject to German data privacy regulations which are considered to be amongst the strictest in the world. Consequently, a special emphasis has to be put on guaranteeing data protection. Little is known about the users themselves, apart from basic things such as their browser type, operation system and similar details that the users’ browser reveals. Prior to forwarding requests to the living lab, *plista* filters out sensitive information (e.g., IP addresses), hence anonymizing the data stream.

In order to join the living lab, participants have to provide a server address and port number. Requests will then be forwarded to these servers. While this guarantees that participants keep complete control over their own code, this setting also comes with the drawback that time is spent for sending and receiving these requests. Keeping in mind that a key requirement for participating in this living lab is that recommendations have to be provided within $< 100\text{ms}$, this network latency can be a serious factor. As we observed in the course of the benchmarking events, this is in particular relevant for teams that are physically remote from the servers in Berlin. In order to address this issue, *plista* provides virtual machines on their server, i.e., participants can participate without the disadvantage of their own network connectivity.

Another challenge when setting up a living lab for the evaluation of information access systems is to thoroughly define the benchmarking metrics and to define the evaluation scenario. Evaluating recommender systems’ performance is subject to intense debates. A variety of evaluation criteria has been defined including rating prediction accuracy, classification, and ranking metrics [16, 30]. The choice of evaluation criteria not only depends on the items but also on the feedback users provide. For instance, rating prediction accuracy metrics such as root mean squared error (RMSE) require users to express their preferences numerically. In the underlying setting, we only observe users interacting with items or disregarding them. We decided to consider the amount of clicks each algorithm obtains as decisive criteria. During the first benchmarking event, some participants joined the evaluation midway through the challenge, i.e., they processed far smaller requests. Consequently, these teams had no chance to win the actual competition. Nevertheless, we consider this criteria fair as long as all participants receive a comparable number of requests. In order to address this time factor, we defined three separate evaluation periods within *NEWSREEL* which were all evaluated individually.

In the living lab scenario, participants can run their algorithms over a longer period of time. This gives them the opportunity to try out different recommenda-

tion techniques and observe the effect of various parameters on their recommendation performance. An important aspect of a benchmarking campaign, however, is also that participants can compare their own performance with state-of-the-art techniques. In order to provide such reference point, we implemented various baseline recommenders [23] which were constantly running during the competitions. Interestingly enough, the baseline algorithms that have been implemented for the NRS challenge turned out to be the most successful recommenders of the challenge, i.e., no participating team was able to beat their performance with respect to the users' click-through rate. Therefore, we consider them to be the state-of-the-art algorithms of such real-life recommendation scenario.

6 Conclusion

In this paper, we introduced a living lab for the evaluation of news recommendations in real-time and described its application during two benchmarking events. The main purpose of living labs is to evaluate user-centric technologies under realistic conditions and context. In the living lab scenario, we interpret this purpose as the provision of news article recommendations for real users who visit news portals to satisfy their personal information needs. The users' context, i.e., the time, interest and the used device is not defined in a laboratory-style evaluation setting but is provided by the actual users themselves. In other words, users follow their own agenda and face no artificial created limitations and conditions. We argue that this challenge can serve as a guideline for the implementation of living labs for the evaluation of information access systems. In fact, first steps towards the creation of a living lab for the evaluation of information retrieval systems are currently discussed [5].

Acknowledgement

The work leading to these results has received funding (or partial funding) from the Central Innovation Programme for SMEs of the German Federal Ministry for Economic Affairs and Energy, as well as from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 610594.

References

1. G. Adomavicius and Young Ok Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *Knowledge and Data Engineering*, 24(5):896–911, may 2012.
2. James Allan. Hard track overview in trec 2003: High accuracy retrieval from documents. In *TREC*, pages 24–37, 2003.
3. Xavier Amatriain. Mining large streams of user data for personalized recommendations. *ACM SIGKDD Explorations Newsletter*, 14(2):37, April 2013.

4. Leif Azzopardi and Krisztian Balog. Towards a living lab for information retrieval research and development - a proposal for a living lab for product search tasks. In *CLEF*, pages 26–37, 2011.
5. Krisztian Balog, David Elsweiler, Evangelos Kanoulas, Liadh Kelly, and Mark Smucker. Report on the ckm workshop on living labs for information retrieval evaluation. *SIGIR Forum*, 48(1), 2014.
6. Nicholas J. Belkin. Some(what) grand challenges for information retrieval. In *ECIR*, page 1, 2008.
7. James Bennett and Stan Lanning. The netflix prize. In *KDDCup’07*, 2007.
8. Torben Brodt and Frank Hopfgartner. Shedding Light on a Living Lab: The CLEF NEWSREEL Open Recommendation Platform. In *Proceedings of the Information Interaction in Context conference, IiX’14*. Springer-Verlag, 2014. to appear.
9. Cyril Cleverdon, Jack Mills, and Michael Keen. Factors determining the performance of indexing systems. Technical report, ASLIB Cranfield project, Cranfield, 1966.
10. Paul Clough and Mark Sanderson. Evaluating the performance of information retrieval systems using test collections. *Information Research*, 18(2), 2013.
11. Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. The Yahoo! Music Dataset and KDD-Cup. In *JMLR: Workshop and Conference Proceedings*, pages 3–18, 2012.
12. Susan Dumais and Nickolas Belkin. The trec interactive tracks: Putting the user into search. In *TREC*, 2005.
13. Cagdas Esiyok, Benjamin Kille, Brijnesh Johannes Jain, Frank Hopfgartner, and Sahin Albayrak. Users’ reading habits in online news portals. In *IiX’14: Proceedings of Information Interaction in Context Conference*. ACM, 08 2014. to appear.
14. Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.
15. Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR ’99, pages 230–237. ACM, 1999.
16. Jonathan L. Herlocker, Joseph a. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.
17. Frank Hopfgartner and Joemon M. Jose. Semantic user profiling techniques for personalised multimedia recommendation. *Multimedia Syst.*, 16(4-5):255–274, 2010.
18. Melody Y. Ivory and Marti A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33(4):470–516, 2001.
19. Jaap Kamps, Shlomo Geva, Carol Peters, Tetsuya Sakai, Andrew Trotman, and Ellen M. Voorhees. Report on the sigir 2009 workshop on the future of ir evaluation. *SIGIR Forum*, 43(2):13–23, 2009.
20. Diane Kelly, Susan T. Dumais, and Jan O. Pedersen. Evaluation challenges and directions for information-seeking support systems. *IEEE Computer*, 42(3):60–66, 2009.
21. Benjamin Kille, Frank Hopfgartner, Torben Brodt, and Tobias Heintz. The plista dataset. In *NRS’13: Proceedings of the International Workshop and Challenge on News Recommender Systems*, pages 14–21. ACM, 10 2013.
22. Joseph Konstan and John Riedl. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1-2):101–123, 2012.

23. Andreas Lommatzsch. Real-time news recommendations using context-aware ensembles. In *ECIR'14: Proceedings of the 36th European conference on Advances in information retrieval*, ECIR'14, pages 51–62. Springer-Verlag, 2014.
24. Andreas Lommatzsch, Till Plumbaum, and Sahin Albayrak. A linked dataverse knows better: Boosting recommendation quality using semantic knowledge. In *Proc. of the 5th Intl. Conf. on Advances in Semantic Processing*, pages 97 – 103, Wilmington, DE, USA, 2011. IARIA.
25. Michael J. Pazzani and Daniel Billsus. Content-based recommendation systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 325–341. Springer Berlin Heidelberg, 2007.
26. Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.
27. Peter Pirolli. Powers of 10: Modeling complex information-seeking systems at multiple scales. *IEEE Computer*, 42(3):33–40, 2009.
28. Alan Said, Jimmy Lin, Alejandro Bellogín, and Arjen de Vries. A month in the life of a production news recommender system. In *Proceedings of the 2013 Workshop on Living Labs for Information Retrieval Evaluation*, LivingLab '13, pages 7–10. ACM, 2013.
29. Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.
30. Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.
31. Mozghan Tavakolifard, Jon Atle Gulla, Kevin C. Almeroth, Frank Hopfgartner, Benjamin Kille, Till Plumbaum, Andreas Lommatzsch, Torben Brodt, Arthur Bucko, and Tobias Heintz. Workshop and challenge on news recommender systems. In *RecSys'13: Proceedings of the International ACM Conference on Recommender Systems*. ACM, 10 2013.
32. TNS Opinion & Social. Special Eurobarometer 386 – Europeans and their Languages. Technical report, European Commission, 2012.
33. David Vallet, Frank Hopfgartner, and Joemon Jose. Use of implicit graph for recommending relevant videos: a simulated evaluation. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 199–210, Berlin, Heidelberg, 2008. Springer-Verlag.
34. Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, MA, USA, 1 edition, 2005.
35. Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *WWW'05*, pages 22–32. ACM, 2005.